PROJECT REPORT

(Project Term June-December 2021)

TESLA SHARE PRICE PREDICTION

Submitted by

Navneet Kumar 11917452

INT 246

(B. Tech CSE)

Under the Guidance of

Sagar Pande

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

DECLARATION

I hereby declare that the project work entitled Tesla Share Price Prediction is an authentic record of my own work carried out as requirements of Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Sagar Pande, during August to November 2021. All the information furnished in this project report is based on my own intensive work and is genuine.

Name of Student: Navneet Kumar
Registration Number: 11917452

CERTIFICATE

This is to certify that the declaration statement made by me is correct to the best of my knowledge and belief. I have completed this Project with sincere efforts. The present work is the result of my original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Name of the Mentor: Sagar Pande

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

ACKNOWLEDGEMENT

I take this opportunity to express my deep gratitude and most sincere thanks to my teacher, parents, and friends for giving most valuable suggestion, helpful guidance, and encouragement in the execution of this project work. I would like to thank our course mentor for guiding us in making the Project work successful.

1. INTRODUCTION

Share market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

I have predicted share price of Tesla of time frame of 10 years. I made use of ANN algorithm to predict the future share price and could differentiate between actual and predicted share price of Tesla share from 2007 to 2017.

1.1 SHARE MARKET PREDICTION

A share market prediction is developed to demonstrate the application of the proposed method. The system follows the dataset, and it consists of two stages, namely, training and testing stages. Training stage trains the dataset and testing stage tests the desired result or output. It is based on Artificial Neural Network algorithm.

1.2 APPLICATION

- Used extensively in share market
- Used by investors and traders in share market
- Used by economic experts for the growth of company and nation
- Used to predict inflation or economic depression.

2. Algorithm Used

Artificial Neural networks (ANN) or neural networks are computational algorithms. It intended to simulate the behaviour of biological systems composed of “neurons”. ANNs are computational models inspired by an animal’s central nervous systems. It is capable of machine learning as well as pattern recognition. These presented as systems of interconnected “neurons” which can compute values from inputs.

A neural network is an oriented graph. It consists of nodes which in the biological analogy represent neurons, connected by arcs. It corresponds to dendrites and synapses. Each arc associated with a weight while at each node. Apply the values received as input by the node and define Activation function along the incoming arcs, adjusted by the weights of the arcs.

A neural network is a machine learning algorithm based on the model of a human neuron. The human brain consists of millions of neurons. It sends and process signals in the form of electrical and chemical signals.

These neurons are connected with a special structure known as synapses. Synapses allow neurons to pass signals. From large numbers of simulated neurons neural networks forms.

An Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large of connected processing units that work together to process information. They also generate meaningful results from it.

We can apply Neural network not only for classification. It can also apply for regression of continuous target attributes.

Neural networks find great application in data mining used in sectors. For example economics, forensics, etc and for pattern recognition. It can be also used for data classification in a large amount of data after careful training.

A neural network may contain the following 3 layers:

- Input layer – The activity of the input units represents the raw information that can feed into the network.
- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.
- Output layer – The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

Artificial Neural Network Layers

Artificial Neural network is typically organized in layers. Layers are being made up of many interconnected ‘nodes’ which contain an ‘activation function’. A neural network may contain the following 3 layers:

a. Input layer

The purpose of the input layer is to receive as input the values of the explanatory attributes for each observation. Usually, the number of input nodes in an input layer is equal to the number of explanatory variables. ‘input layer’ presents the patterns to the network, which communicates to one or more ‘hidden layers’.

The nodes of the input layer are passive, meaning they do not change the data. They receive a single value on their input and duplicate the value to their many outputs. From the input layer, it duplicates each value and sent to all the hidden nodes.

b. Hidden layer

The Hidden layers apply given transformations to the input values inside the network. In this, incoming arcs that go from other hidden nodes or from input nodes connected to each node. It connects with outgoing arcs to output nodes or to other hidden nodes. In hidden layer, the actual processing is done via a system of weighted ‘connections’. There may be one or more hidden layers. The values entering a hidden node multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number.

c. Output layer

The hidden layers then link to an ‘output layer‘. Output layer receives connections from hidden layers or from input layer. It returns an output value that corresponds to the prediction of the response variable. In classification problems, there is usually only one output node. The active nodes of the output layer combine and change the data to produce the output values.

The ability of the neural network to provide useful data manipulation lies in the proper selection of the weights. This is different from conventional information processing.

Structure of a Neural Network

The structure of a neural network also referred to as its ‘architecture’ or ‘topology’. It consists of the number of layers, Elementary units. It also consists of Interchanged Weight adjustment mechanism. The choice of the structure determines the results which are going to obtain. It is the most critical part of the implementation of a neural network.

The simplest structure is the one in which units distributes in two layers: An input layer and an output layer. Each unit in the input layer has a single input and a single output which is equal to the input. The output unit has all the units of the input layer connected to its input, with a combination function and a transfer function. There may be more than 1 output unit. In this case, resulting model is a linear or logistic regression. This is depending on whether transfer function is linear or logistic. The weights of the network are regression coefficients.

By adding 1 or more hidden layers between the input and output layers and units in this layer the predictive power of neural network increases. But a number of hidden layers should be as small as possible. This ensures that neural network does not store all information from learning set but can generalize it to avoid overfitting.

Overfitting can occur. It occurs when weights make the system learn details of learning set instead of discovering structures. This happens when size of learning set is too small in relation to the complexity of the model.

A hidden layer is present or not, the output layer of the network can sometimes have many units, when there are many classes to predict.

Advantages and Disadvantages of Neural Networks

Let us see few advantages and disadvantages of neural networks:

- Neural networks perform well with linear and nonlinear data but a common criticism of neural networks, particularly in robotics, is that they require a large diversity of training for real-world operation. This is so because any learning machine needs sufficient representative examples in order to capture the underlying structure that allows it to generalize to new cases.
- Neural networks works even if one or few units fail to respond to network but to implement large and effective software neural networks, much processing and storage resources need to be committed. While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a most simplified form on Von Neumann technology may compel a neural network designer to fill millions of database rows for its connections – which can consume vast amounts of computer memory and hard disk space.
- Neural network learns from the analyzed data and does not require to reprogramming but they are referred to as black box" models, and provide very little insight into what these models really do. The user just needs to feed it input and watch it train and await the output.

3. Libraries Used

3.1 Numpy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

3.2 Pandas

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

3.3 Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with datframes and pandas library.The graphs created can also be customized easily.

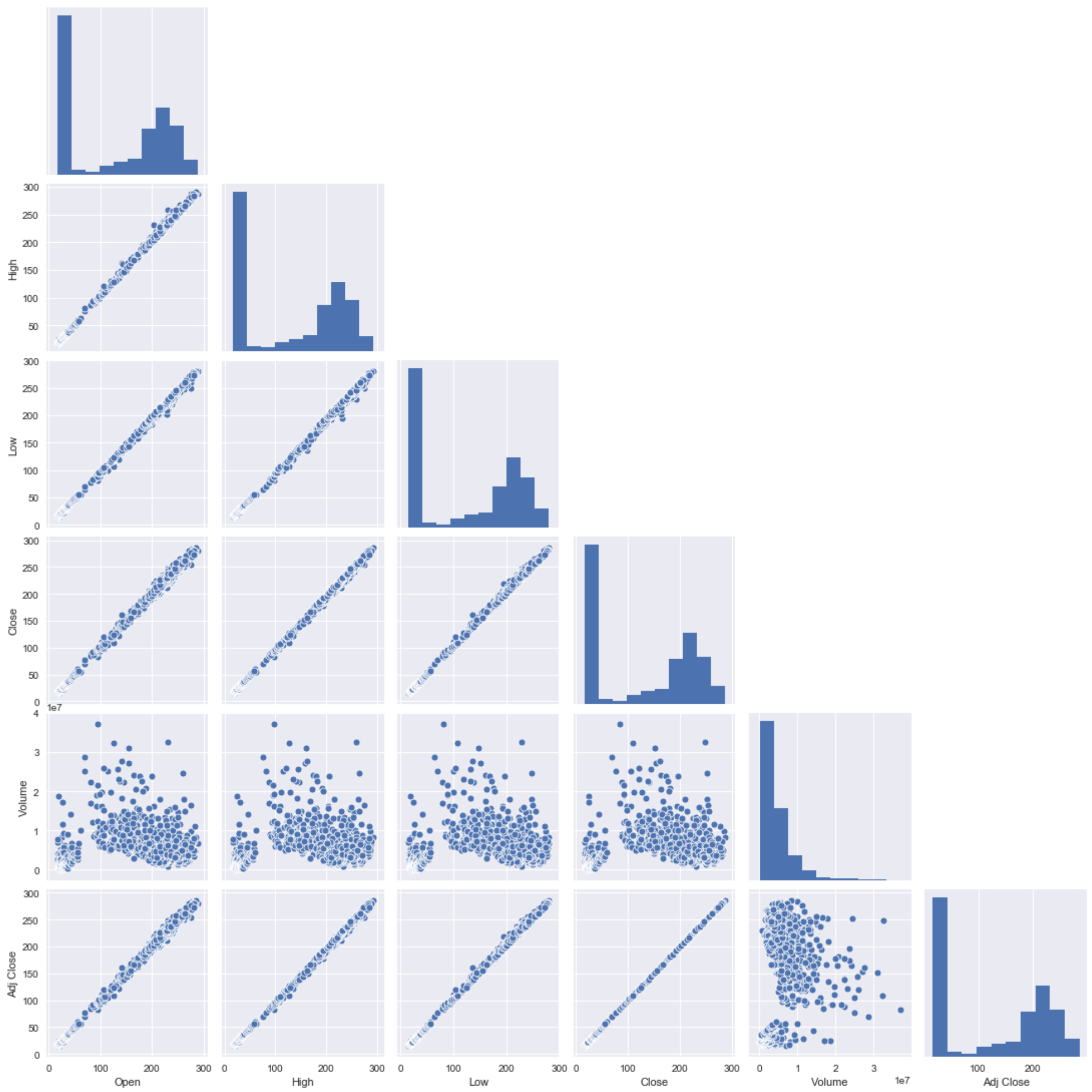
3.4 TensorFlow

TensorFlow is a free and open source library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural network**.**

Source Code

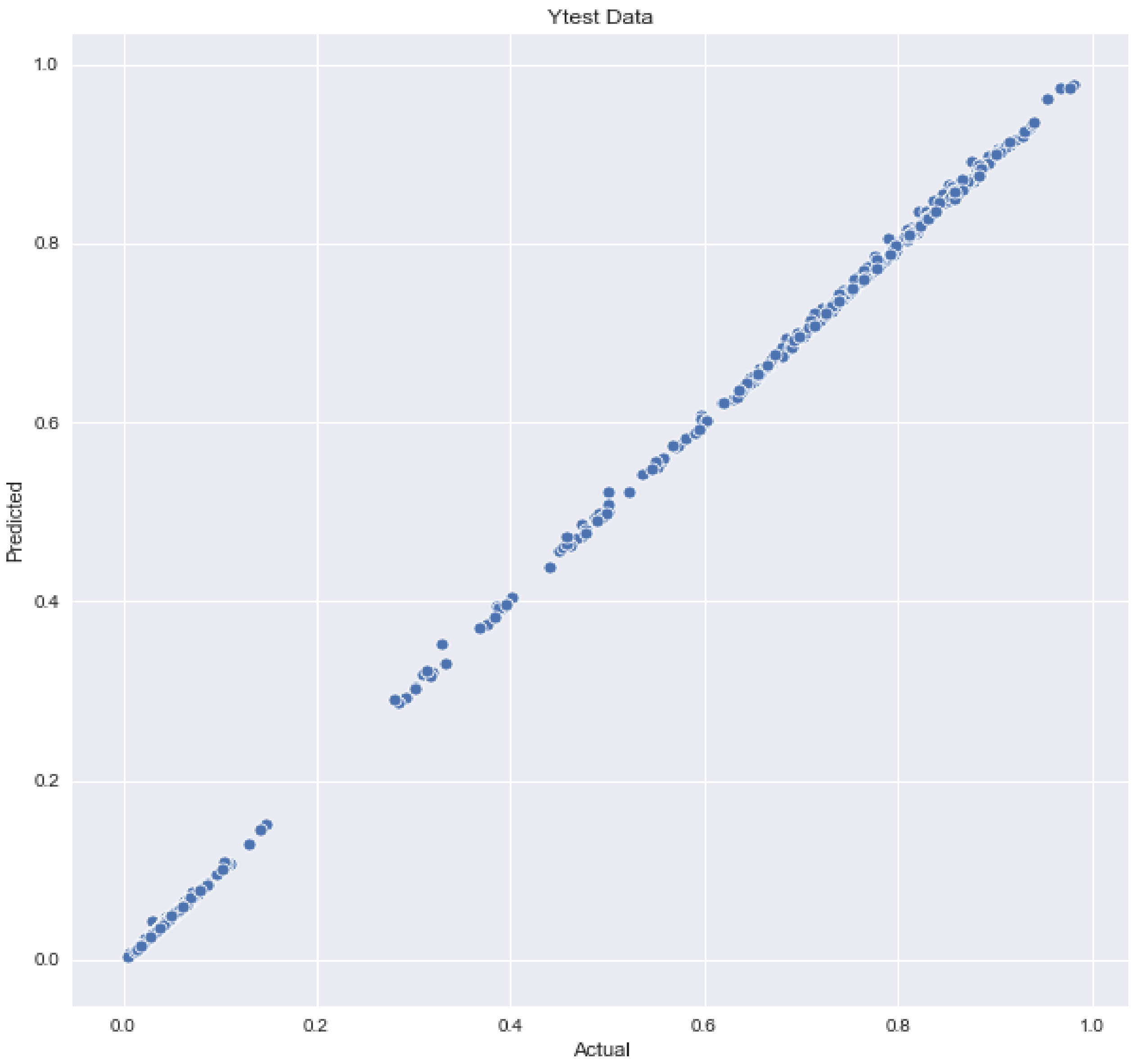
```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
plt.style.use('seaborn')

from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
%matplotlib inline


df=pd.read_csv("Tesla.csv")
df.shape

plt.figure(figsize=(10,10))
sns.pairplot(data=df,kind='scatter',corner=True)
plt.show()
```

```
<Figure size 720x720 with 0 Axes>
```

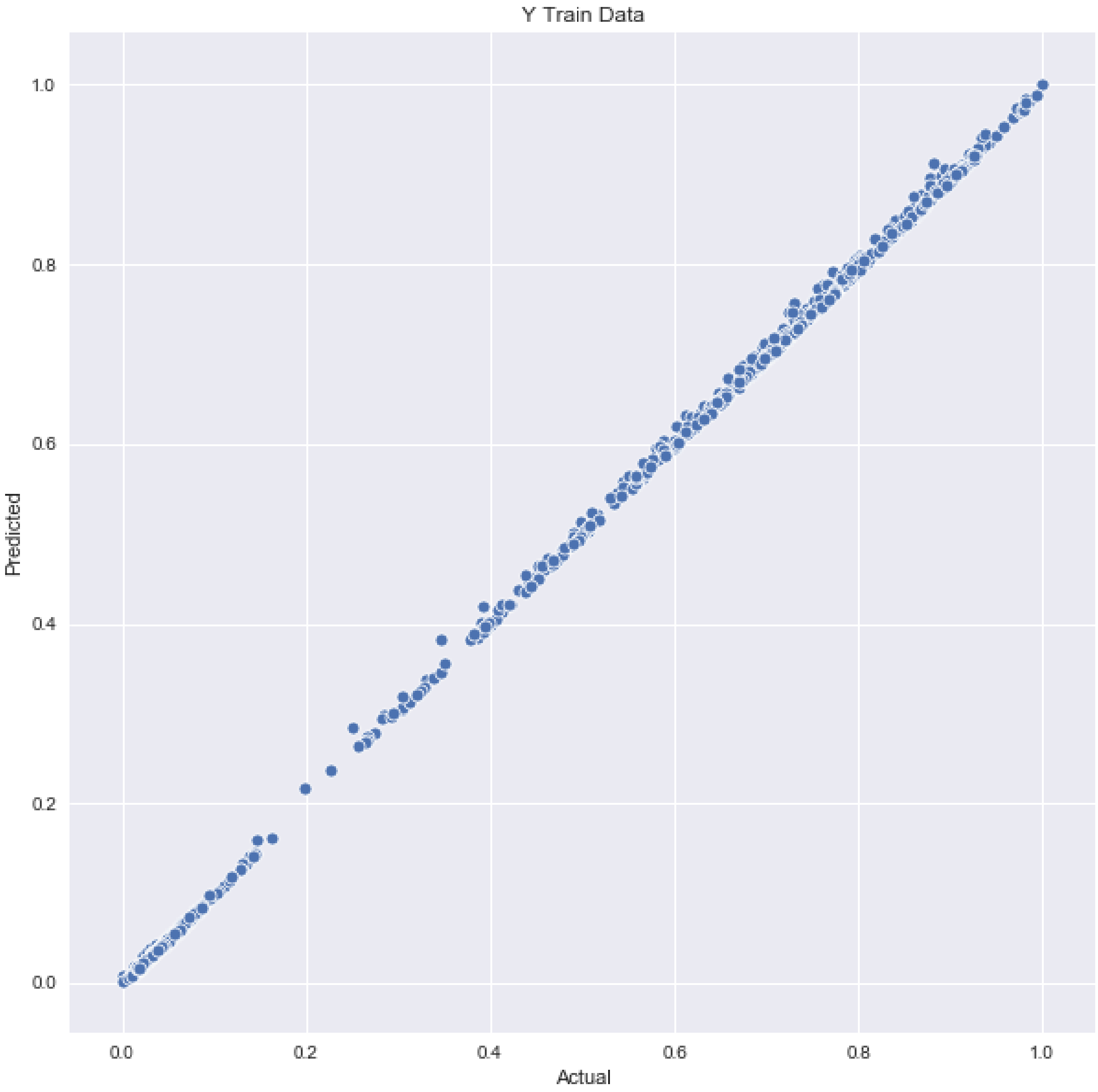


```
scale=MinMaxScaler()
scalled=scale.fit_transform(df)
i=0
for col in df.columns:
    df[col]=scalled[:,i]
    i+=1
x,y= df.drop('Adj Close',axis=1), df['Adj Close']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
x_train.shape,x_test.shape
```

```
model=keras.Sequential([
    keras.layers.Dense(5, input_shape=(5,), activation='linear'),
    keras.layers.Dense(1,activation='linear')
])

model.compile(
loss='mean_squared_error',
optimizer='adam')
model.fit(x_train,y_train,epochs=50)
model.evaluate(x_test,y_test)
y_pred_test=model.predict(x_test).flatten()
y_pred_train=model.predict(x_train).flatten()
def actual_vs_predicted(y_true,y_pred,title=None,xscale='linear',
                       yscale='linear',return_correlation=False):
    x='Actual'
    y='Predicted'
    plt.figure(figsize=(10,10))
    ap=pd.DataFrame({x:y_true,y:y_pred})
    sns.scatterplot(data=ap,x=x,y=y)
    plt.title(title)
    plt.xscale(xscale)
    plt.yscale(yscale)
    plt.show()
    if return_correlation:
        return ap.corr()
    actual_vs_predicted(y_test,y_pred_test,title="Ytest Data",
                   return_correlation=True)
```

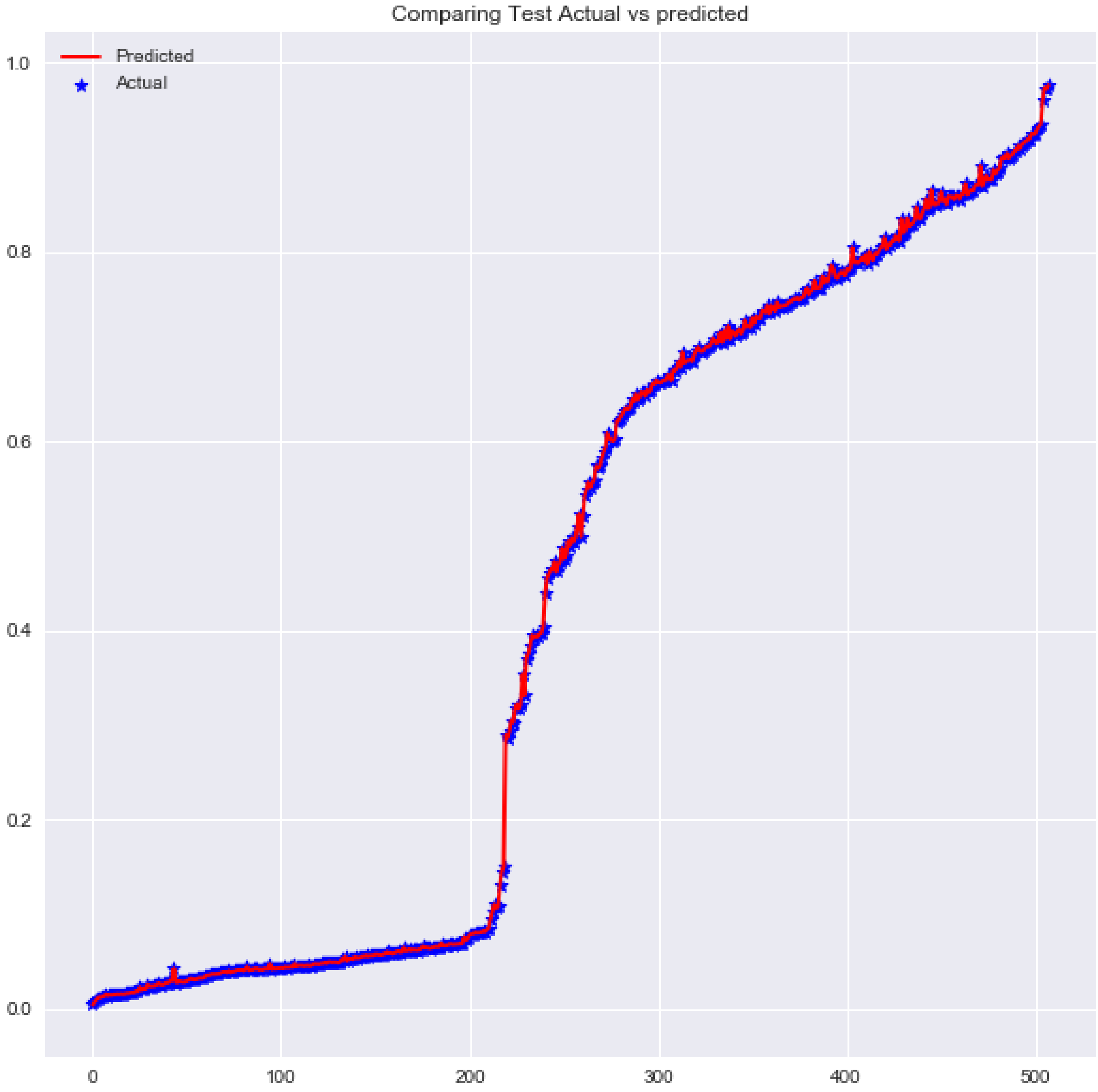


```
r2_score(y_pred_test,y_test)

actual_vs_predicted(y_train,y_pred_train,title="Y Train Data")
```

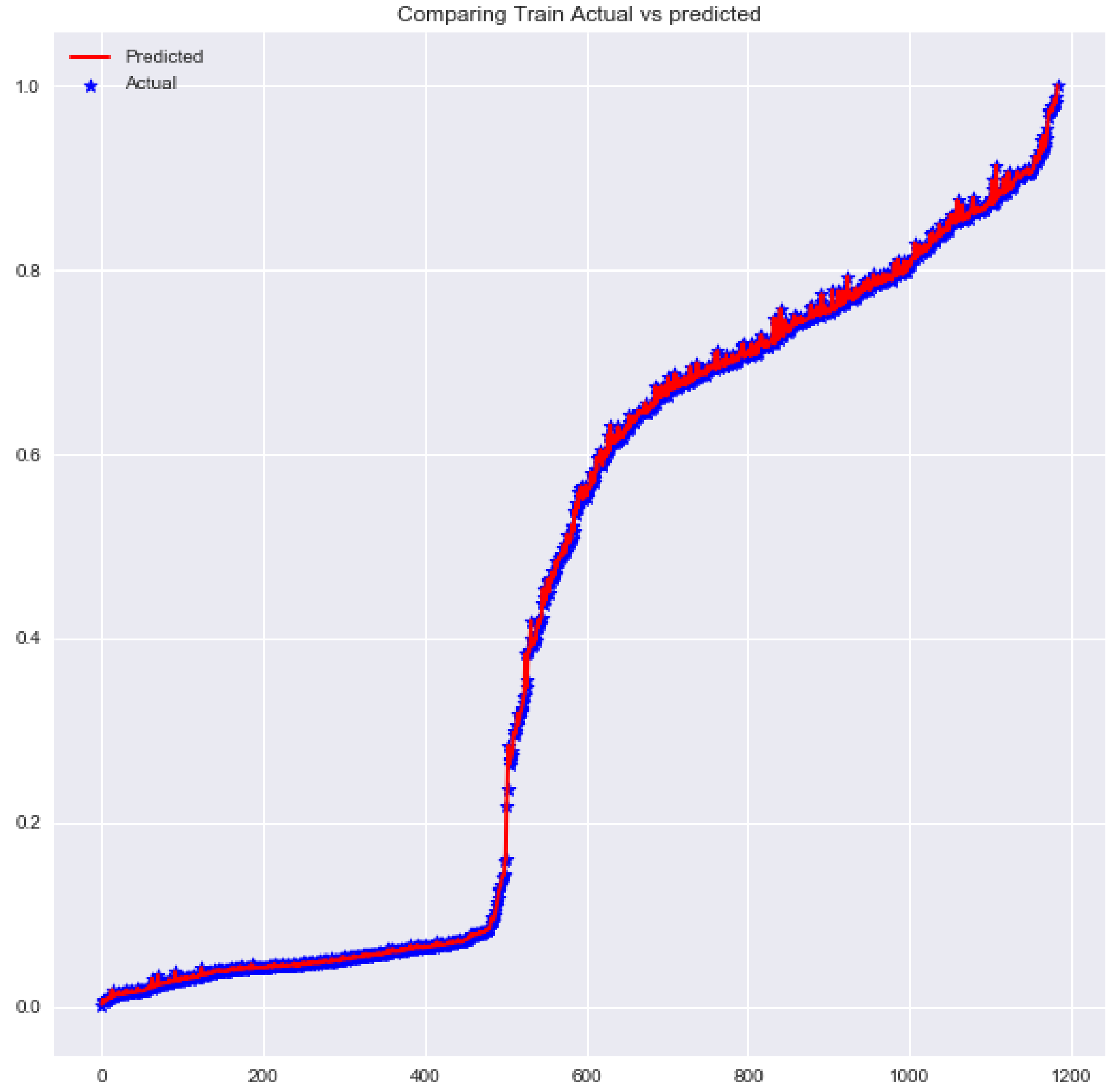


```
        def plot_actual_vs_predicted(y_true,y_pred,title=None,xscale='linear',
         yscale='linear'):
x='Actual'
y='Predicted'
plt.figure(figsize=(10,10))
ap=pd.DataFrame({x:y_true,y:y_pred})
ap.sort_values(by=x,ascending=True,inplace=True)
```

```
plt.scatter(np.arange(len(ap[x])),ap[y],color='blue',label=x,linewidth=0.9,marker='*')

plt.plot(np.arange(len(ap[x])),ap[y],color='red',label=y,linewidth=2)


plt.title(title)

plt.legend()

plt.xscale(xscale)

plt.yscale(yscale)

plt.show()

plot_actual_vs_predicted(y_test,y_pred_test,"Comparing Test Actual vs predicted")
```



```
plot_actual_vs_predicted(y_train,y_pred_train,"Comparing Train Actual vs predicted")
```



CONCLUSION

Hence, I would like to conclude my report on note that it was a great learning experience for me and my future endeavour. I would also thank my mentor Sagar Pande sir for able guidance and support. I would also incorporate GUI to enhance my project.

BIBLIOGRAPHY

www.kaggle.com

www.geeksforgeeks.com

www.machinelearning.com

www.tensorflow.com